

Patent Abstracts of Japan

PUBLICATION NUMBER : 2001256075
PUBLICATION DATE : 21-09-01

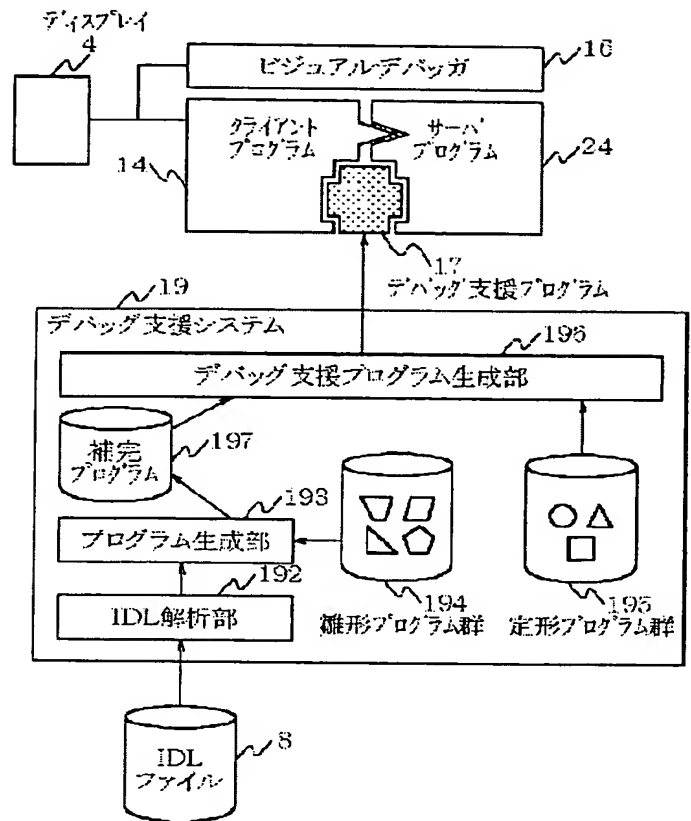
APPLICATION DATE : 13-03-00
APPLICATION NUMBER : 2000068151

APPLICANT : NEC SOFTWARE HOKURIKU LTD;

INVENTOR : SUGAUCHI MASAKAZU;

INT.CL. : G06F 11/28 G06F 9/06 G06F 15/16

TITLE : DEVELOPING SYSTEM AND
DEVELOPING METHOD AND
RECORDING MEDIUM



ABSTRACT : PROBLEM TO BE SOLVED: To solve the problem that it is necessary to realize a proper work to repeat processes for embedding a trace output sentence in a serve program, for performing re-compile link and an application, and for confirming a trace- outputted file step by step in debugging a server program.

SOLUTION: An IDL analyzing part 192 analyzes the meaning of interface information defined in an IDL file 8(a step A-2). A program generating part 193 operates the extraction and assembly of a logical logic from a model program group 194, and prepares a complementary program 197 exclusive for the IDL file 8 (a step A-3). A debug supporting program generating part 196 prepares a debug supporting program 17 from the complementary program 197 prepared by the program generating part 193 and a stylized program group 5 (a step A-4). Thus, it is possible to realize the operation of a visual debugger 16 in a client developing environment which is rich in debug functions (a step A-5).

COPYRIGHT: (C)2001,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-256075

(P2001-256075A)

(43) 公開日 平成13年9月21日 (2001.9.21)

(51) Int.Cl. ⁷	識別記号	F I	テマコード (参考)
G 0 6 F 11/28		G 0 6 F 11/28	J 5 B 0 4 2
			P 5 B 0 4 5
9/06	5 3 0	9/06	5 3 0 V 5 B 0 7 6
15/16	6 2 0	15/16	6 2 0 S
	6 3 0		6 3 0 A

審査請求 有 請求項の数 7 O L (全 8 頁)

(21) 出願番号 特願2000-68151(P2000-68151)

(22) 出願日 平成12年3月13日 (2000.3.13)

(71) 出願人 000242666

北陸日本電気ソフトウェア株式会社

石川県石川郡鶴来町安養寺1番地

(72) 発明者 管内 正和

石川県石川郡鶴来町安養寺1番地 北陸日

本電気ソフトウェア株式会社内

(74) 代理人 100082935

弁理士 京本 直樹 (外2名)

Fターム (参考) 5B042 GA12 H111 NN50

5B045 AA00 BB12 BB28 BB48 CC09

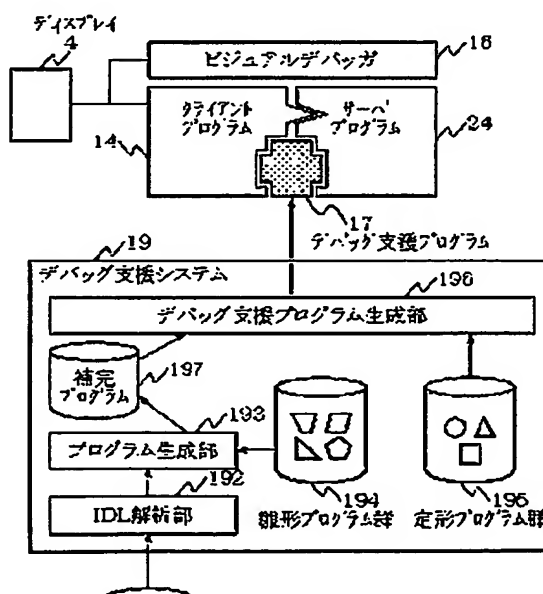
5B076 DD04 EE01

(54) 【発明の名称】 開発システム、開発方法、記録媒体

(57) 【要約】

【課題】 サーバプログラムのデバッグをする場合、サーバプログラムにトレース出力文を埋め込み、再コンパイル・リンク、アプリケーション実行、トレース出力されたファイルの確認という工程を繰り返す地道な作業を必要とした。

【解決手段】 IDL解析部192はIDLファイル8に定義されているインタフェース情報の意味解析を行う (ステップA-2)。プログラム生成部193は雛形プログラム群194より処理ロジックの抽出と組み立てを行い、IDLファイル8専用の縮完プログラム197を作成する (ステップA-3)。デバッグ支援プログラム生成部196はプログラム生成部193が作成した縮完プログラム197と定形プログラム群196より、デバッグ支援プログラム17を作成する (ステップA-4)。デバッグ機能が豊富なクライアント開発環境でのビジュアルデバッグ16の動作を可能にする (ステップA-



(2)

特開2001-256075

1

2

【特許請求の範囲】

【請求項1】 同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、前記クライアントはビジュアルなデバッグ環境を有し、前記サーバ上で実行されるサーバプログラムを前記クライアント上で実行させることにより、前記クライアントが有する前記ビジュアルなデバッグ環境を利用することを特徴とする開発システム。

【請求項2】 同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、

前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムと前記クライアントプログラムと前記サーバプログラムを接続するORB (Object Request Broker) から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、前記デバッグ支援システムは、IDL (Interface Definition Language) ファイルを読み込み解析するIDL解析部と前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成するプログラム生成部と前記縮完プログラムと定形プログラム群からデバッグ支援プログラムを生成するデバッグ支援プログラム生成部から構成され、

前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムを前記クライアント上で実行させることにより、前記クライアントが有する前記ビジュアルなデバッグ環境を利用することを特徴とする開発システム。

【請求項3】 同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムとミドルウェアAPI群と前記クライアントプログラムと前記サーバプログラムを接続するORB (Object Request Broker) から構成され、

前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、前記デバッグ支援システムは、IDL (Interface Definition Language) ファイルを読み込み解析するIDL解析部と、前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成するプログラム生成部と、前記縮完プログラムと定形プログラム群と前記ミドルウェアAPIプログラム群からデバッグ支援プログラムを生成するデバッグ支援プログラム生成部から構成され、

ビジュアルなデバッグ環境を利用することを特徴とする開発システム。

【請求項4】 同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発方法であって、

前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムと前記クライアントプログラムと前記サーバプログラムを接続するORB (Object Request Broker) から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、

IDL (Interface Definition Language) ファイルを読み込み解析する第一のステップと前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成する第二のステップと前記縮完プログラムと定形プログラム群からデバッグ支援プログラムを生成する第三のステップと、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムを前記クライアント上で実行させる第四のステップから構成されることを特徴とする開発方法。

【請求項5】 同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発方法であって、

前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムとミドルウェアAPI群と前記クライアントプログラムと前記サーバプログラムを接続するORB (Object Request Broker) から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、

IDL (Interface Definition Language) ファイルを読み込み解析する第一のステップと、前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成する第二のステップと、前記縮完プログラムと定形プログラム群と前記ミドルウェアAPIプログラム群からデバッグ支援プログラムを生成する第三のステップと、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムを前記クライアント上で実行させる第四のステップから構成されることを特徴とする開発方法。

【請求項6】 請求項4記載の第一乃至第四のステップをコンピュータに実行させるプログラムを記録したことを特徴とする記録媒体。

【請求項7】 請求項5記載の第一乃至第四のステップをコンピュータに実行させるプログラムを記録したことを特徴とする記録媒体。

(3)

特開2001-256075

3

ンプログラムの開発に関し、特にサーバプログラムの開発方式、開発方法に関する。

【0002】

【従来の技術】従来、分散アプリケーションを開発する手段として、オブジェクト指向とクライアント、サーバを融合した分散オブジェクト技術であるCORBAを用いる場合、開発の手順としてIDL（インターフェース定義言語）の定義、IDLコンパイラの起動、生成されたスタブ、スケルトンを用いて、クライアントプログラム、サーバプログラムのコンパイル・リンクを行っていた。

【0003】特開平6-301526号公報「クライアント／サーバモデルのプログラム開発支援方式」には、インタフェース定義言語IDLで記述されたインタフェース定義を作成するとき、あるいはクライアント／サーバモデルのプログラムからリモートプロシージャコールRPCのライブラリを読み出すとき、クライアント／サーバモデルのプログラムの記述言語を意識する必要のないクライアント／サーバモデルのプログラム開発支援方式が開示されている。

【0004】

【発明が解決しようとする課題】従来、このような分散アプリケーションにおいて、サーバ上で実行されるサーバプログラムのデバッグを実施する場合は、サーバプログラムにトレース出力文をひたすら埋め込み、再コンパイル・リンク、アプリケーション実行、トレース出力されたファイルの確認という工程を繰り返さなくてはならないという問題があった。

【0005】特開平6-301526号公報「クライアント／サーバモデルのプログラム開発支援方式」は、クライアント開発環境が用意するビジュアルなデバッグ機能を利用することは開示されていない。

【0006】本発明の目的は、分散オブジェクト技術であるCORBA（Common Object Request Broker Architecture）を利用する分散オブジェクトアプリケーションにおいて、本来はサーバ上で実行されるサーバプログラムをクライアント上で実行させることにより、クライアント開発環境が用意するビジュアルなデバッグ機能を利用可能にする構成を提供することである。

【0007】

【課題を解決するための手段】本発明第一の開発システムは、同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、前記クライアントはビジュアルなデバッグ環境を有し、前記サーバ上で実行されるサーバプログラムを前記クライアント上で実行させることにより、前記ク

4

セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムと前記クライアントプログラムと前記サーバプログラムを接続するORB（Object Request Broker）から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、前記デバッグ支援システムは、IDL（Interface Definition Language）ファイルを読み込み解析するIDL解析部と前記IDL解析部の結果と雛形プログラム群から縮充プログラムを生成するプログラム生成部と前記縮充プログラムと定形プログラム群からデバッグ支援プログラムを生成するデバッグ支援プログラム生成部から構成され、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムと前記クライアント上で実行させることにより、前記クライアントが有する前記ビジュアルなデバッグ環境を利用する。

【0009】本発明第三の開発システムは、同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発システムであって、前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムとミドルウェアAPI群と前記クライアントプログラムと前記サーバプログラムを接続するORB（Object Request Broker）から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、前記デバッグ支援システムは、IDL（Interface Definition Language）ファイルを読み込み解析するIDL解析部と、前記IDL解析部の結果と雛形プログラム群から縮充プログラムを生成するプログラム生成部と、前記縮充プログラムと定形プログラム群と前記ミドルウェアAPIプログラム群からデバッグ支援プログラムを生成するデバッグ支援プログラム生成部から構成され、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムと前記クライアント上で実行させることにより、前記クライアントが有する前記ビジュアルなデバッグ環境を利用する。

【0010】本発明第一の開発方法は、同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発方法であって、前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムと前記クライアントプログラムと前記サーバプログラムを接続するORB（Object R

(4)

特開2001-256075

5

6

ルを読み込み解析する第一のステップと前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成する第二のステップと前記縮完プログラムと定形プログラム群からデバッグ支援プログラムを生成する第三のステップと、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムを前記クライアント上で実行させる第四のステップから構成される。

【0011】本発明第二の開発方法は、同一の命令セットを有するクライアントとサーバをネットワークにより接続したシステムにおける分散オブジェクトアプリケーションプログラムの開発方法であって、前記分散オブジェクトアプリケーションプログラムはクライアントプログラムとサーバプログラムとミドルウェアAPI群と前記クライアントプログラムと前記サーバプログラムを接続するORB (Object Request Broker) から構成され、前記クライアントはビジュアルなデバッグ環境と、デバッグ支援システムとを有し、IDL (Interface Definition Language) ファイルを読み込み解析する第一のステップと、前記IDL解析部の結果と雛形プログラム群から縮完プログラムを生成する第二のステップと、前記縮完プログラムと定形プログラム群と前記ミドルウェアAPIプログラム群からデバッグ支援プログラムを生成する第三のステップと、前記デバッグ支援プログラムと前記サーバプログラムと前記クライアントプログラムを前記クライアント上で実行させる第四のステップから構成される。

【0012】本発明第一の記録媒体は、本発明第一の開発方法第一乃至第四のステップをコンピュータに実行させるプログラムを記録した。

【0013】本発明第二の記録媒体は、本発明第二の開発方法第一乃至第四のステップをコンピュータに実行させるプログラムを記録した。

【0014】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して詳細に説明する。図1は、本発明の実施の形態の開発方法の概略を示すブロック図である。クライアント1とサーバ2はネットワーク3により接続されている。クライアント1はディスプレイ部4が接続されビジュアルな環境を備えている。サーバ2はディスプレイ部4を有しておらずビジュアルな環境を保有していない。クライアントオブジェクト11はクライアント1で動作し、サーバオブジェクト21はサーバ上で動作する。クライアントオブジェクト11とサーバオブジェクト21は同一の命令セットを有するコンピュータ、つまりサーバ2またはクライアント1で動作する。サーバオブジェクト21を開発するために、本発明では、サーバオブジェクト21をクライアント1上で動作させ、クラ

イアント1上に用意される。

【0015】次に、本発明の実施の形態の動作について図面を参照して説明する。まず、本発明のシステムの実行環境について説明する。図2は、本発明第一の実施の形態のデバッグ支援システム19のインタフェースを示すブロック図である。クライアントオブジェクト11またはサーバオブジェクト21である分散オブジェクトの呼び出しは以下のようにして行われる。分散オブジェクトのインタフェースが定義されているIDL (Interface Definition Language) ファイル8は、IDLコンパイラ12により、クライアントスタブ13が作成される。また、IDLコンパイラ22によりとサーバスケルトン23が作成される。クライアントプログラム14はクライアントスタブ13を利用してサーバ2上のサーバオブジェクト21を呼び出す。クライアントスタブ13はORB (Object Request Broker) 15、ORB25を経由してサーバスケルトン23を呼び出すことができる。サーバスケルトン23よりサーバプログラム24が動作する。

【0016】さらに、本発明のデバッグ環境について説明する。IDLファイル8を取り込みデバッグ支援プログラム17を生成するデバッグ支援システム19。クライアントプログラム14、サーバプログラム24、デバッグ支援プログラム17とビジュアルデバッガ16から構成される。ビジュアルデバッガ16は、ビジュアルなデバッグ機能を有し、画面からのトレースの指示、ブレークポイントの設定、コンパイル、リンク等デバッグに必要な機能を装備し、開発者とのインタフェースとしてビジュアルな環境を保有する。また、本発明のデバッグ環境では、クライアントスタブ13およびサーバスケルトン23は存在せず、ミドルウェアレスという非常にシンプルな環境となる。

【0017】次に本発明第一の実施の形態のデバッグシステムの動作の説明をする。クライアントプログラム14は、サーバ2上に配置されている本来のサーバオブジェクト21を呼び出すことはしない。クライアントプログラム14があるクライアント1上に移行されているサーバプログラム24を直接呼び出す。

【0018】基本的な考え方は説明した通りであるが、分散オブジェクトとして作成されたクライアントプログラム14とサーバプログラム24は、CORBAが提供するインタフェースを使用しているため、この2つのプログラムだけでアプリケーションとして動作させることはできない。具体的に説明すると、ORB15の初期化・終了などのインタフェース、サーバオブジェクトのリファレンス取得・解放などのインタフェース、シーケンスオブジェクトの要素参照・設定などのインタフェー

(5)

特開2001-256075

7

台。これらの必要となるインタフェースを補完しなければアプリケーションとして動作させることができない。デバッグ支援システム19は、この補完プログラム197を自動生成する。

【0019】次に、図3と図4を用いてデバッグ支援システム19を中心とした動作の説明をする。図3は、本発明第一の実施の形態のデバッグ支援システム19の内部構成をブロック図である。図4は、本発明第一の実施の形態の開発方法の動作を示すフローチャートである。

【0020】デバッグ支援システム19では、まず、IDLファイル8を取り込む(ステップA-1)。次に、IDL解析部192はIDLファイル8に定義されているインタフェース情報の意味解析を行う(ステップA-2)。IDLファイル8にはクライアントプログラム14とサーバプログラム24が利用する全てのインタフェース情報が定義されているので、意味解析を行うことによりアプリケーションとして動作させるために必要となる補完プログラム197を決定できる。プログラム生成部193は雛形プログラム群194より処理ロジックの抽出と組み立てを行い、IDLファイル8専用の補完プログラム197を作成する(ステップA-3)。雛形プログラム群194には、IDLコンパイラ12、22が作成するシーケンスオブジェクトのインタフェースや遅延同期のインタフェースなどを補完するために必要な部品ロジックを格納している。デバッグ支援プログラム生成部196はプログラム生成部193が作成した補完プログラム197と定形プログラム群195より、デバッグ支援プログラム17を作成する(ステップA-4)。定形プログラム群195には、分散オブジェクトアプリケーションにとって必須記述となるORB15の初期化・終了、サーバオブジェクト21のリファレンス取得などの定形ロジックを格納している。

【0021】このようにして、デバッグ支援プログラム17は、分散オブジェクトとして記述されているクライアントプログラム14とサーバプログラム24をミドルウェアレスという環境で、デバッグ機能が豊富なクライアント開発環境でのビジュアルデバッガ16でのデバッグを行う(ステップA-5)。

【0022】次に、クライアントプログラム14およびサーバプログラム24が、実装製品から提供されているミドルウェアAPI26、36を使用している場合について、説明する。分散オブジェクトを実現するCORBA実装製品では、アプリケーション管理および制御、ノード管理やメモリ管理などを行うために、さまざまなミドルウェアAPI(アプリケーションインタフェース)26、36を提供している。図5は、本発明第二の実施の形態のデバッグ支援システム19bのインタフェースを

8

ム24が、実装製品から提供されているミドルウェアAPI36、ミドルウェアAPI26を使用している場合を説明する。デバッグ支援システムb19bで、デバッグ支援プログラムb17bの作成において、雛形プログラム群194と定形プログラム群195と、さらにミドルウェアAPIプログラム群198を使用する。このミドルウェアAPIプログラム群198は、実装製品別に用意し、対象となるプログラム群が抽出されてデバッグ支援プログラムb17bとなる。

【0023】デバッグ支援システムb19bでは、まず、IDLファイル8を取り込む(ステップB-1)。次に、IDL解析部192はIDLファイル8に定義されているインタフェース情報の意味解析を行う(ステップB-2)。IDLファイル8にはクライアントプログラム14とサーバプログラム24が利用する全てのインタフェース情報が定義されているので、意味解析を行うことによりアプリケーションとして動作させるために必要となる補完プログラム197を決定できる。プログラム生成部193は雛形プログラム群194より処理ロジックの抽出と組み立てを行い、IDLファイル8専用の補完プログラム197を作成する(ステップB-3)。雛形プログラム群194には、IDLコンパイラ12、22が作成するシーケンスオブジェクトのインタフェースや遅延同期のインタフェースなどを補完するために必要な部品ロジックを格納している。ミドルウェアAPIプログラム群198は、実装製品別に用意し、対象となるプログラム群を抽出する(ステップB-4)。デバッグ支援プログラム生成部b196bはプログラム生成部193が作成した補完プログラム197と定形プログラム群195とミドルウェアAPIプログラム群198より、デバッグ支援プログラムb17bを作成する(ステップB-4)。定形プログラム群195には、分散オブジェクトアプリケーションにとって必須記述となるORB15の初期化・終了、サーバオブジェクト21のリファレンス取得などの定形ロジックを格納している。このようにして、デバッグ機能が豊富なクライアント開発環境でのビジュアルデバッガ16の動作を可能にする(ステップB-6)。

【0024】以上のように、このミドルウェアAPIプログラム群198の付加により、本発明のデバッグ支援システムは、あらゆる分散オブジェクトアプリケーションの開発支援が可能となる。

【0025】上述のA-1からA-5の各ステップとB-1からB-6の各ステップはコンピュータのプログラムとして実行することが出来る。図7は、本発明実施の形態の開発方法を実行するコンピュータ60と記録媒体61を示す図である。

(6)

特開2001-256075

9

10

クライアントプログラムとサーバプログラムを動作させることができることにある。その理由は、ミドルウェアによる分散オブジェクト環境がない、もしくは設定ができない環境においても、分散オブジェクトアプリケーションの開発が行うことができるためである。

【0027】第2の効果は、本来はサーバ上で実行されるサーバプログラムをクライアント上で実行させることにより、クライアント開発環境が用意するビジュアルなデバッグ機能を利用可能にすることにある。その理由は、サーバマシン上で実行されるプログラムのデバッグを実施するためには、サーバプログラムにトレース出力文を埋め込むなど作業が必要であるが、クライアントマシン上で実行できることにより、ビジュアルなデバッグ製品を利用してデバッグ工数を大幅に削減することができるためである。

【図面の簡単な説明】

【図1】本発明実施の形態の開発方法の概略を示すブロック図である。

【図2】本発明第一の実施の形態のデバッグ支援システム19のインタフェースを示すブロック図である。

【図3】本発明第一の実施の形態のデバッグ支援システム19の内部構成をブロック図である。

【図4】本発明第一の実施の形態の開発方法の動作を示すフローチャートである。

【図5】本発明第二の実施の形態のデバッグ支援システムb19bのインタフェースを示すブロック図である。

【図6】本発明第二の実施の形態の開発方法の動作を示すフローチャートである。

【図7】本発明実施の形態の開発方法を実行するコンピュータと記録媒体を示す図である。

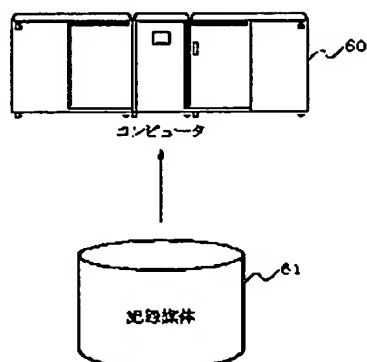
【符号の説明】

1 クライアント

*

* 2	サーバ
3	ネットワーク
4	ディスプレイ部
8	IDLファイル
11	クライアントオブジェクト
12	IDLコンパイラ
13	クライアントスタブ
14	クライアントプログラム
15	ORB
16	ビジュアルデバッガ
17	デバッグ支援プログラム
17 b	デバッグ支援プログラムb
19	デバッグ支援システム
19 b	デバッグ支援システムb
21	サーバオブジェクト
22	IDLコンパイラ
23	サーバスケルトン
24	サーバプログラム
25	ORB
26	ミドルウェアAPI
36	ミドルウェアAPI
60	コンピュータ
61	記録媒体
192	IDL解析部
193	プログラム生成部
194	雛形プログラム群
195	定形プログラム群
196	デバッグ支援プログラム生成部
196 b	デバッグ支援プログラム生成部b
197	縮完プログラム
198	ミドルウェアAPIプログラム群

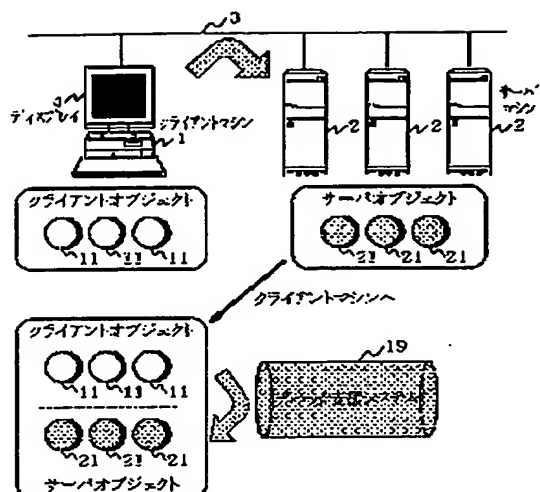
【図7】



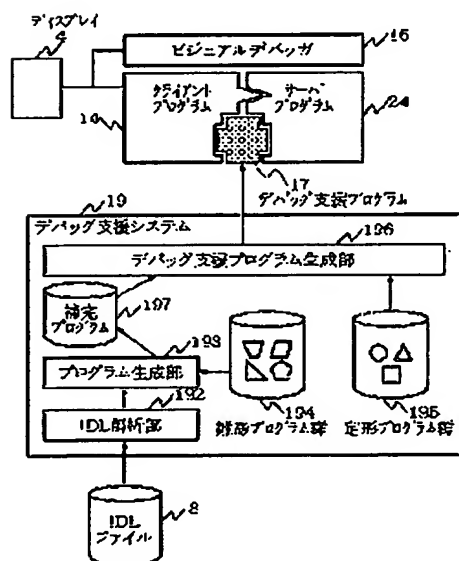
(7)

特開2001-256075

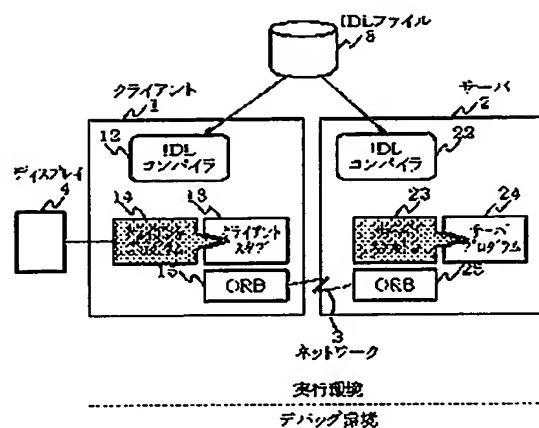
【図1】



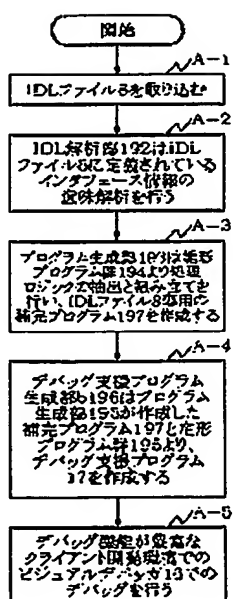
【図3】



【図2】



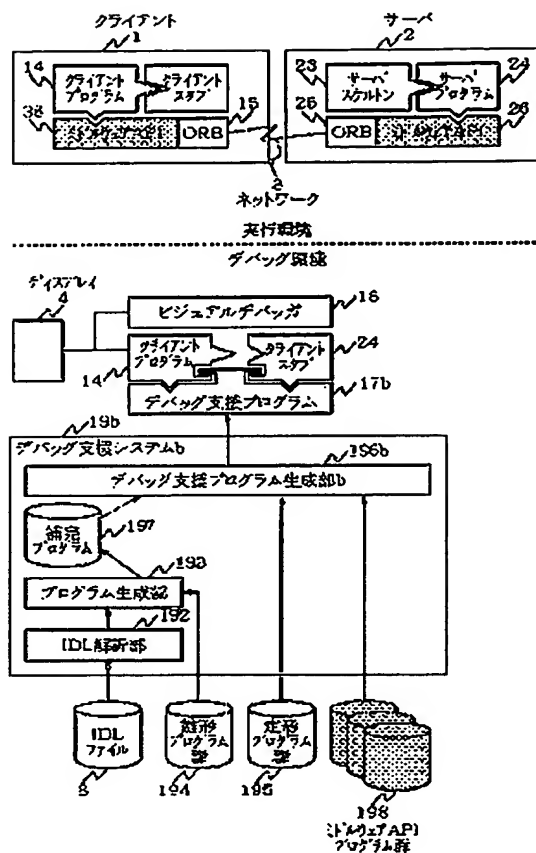
【図4】



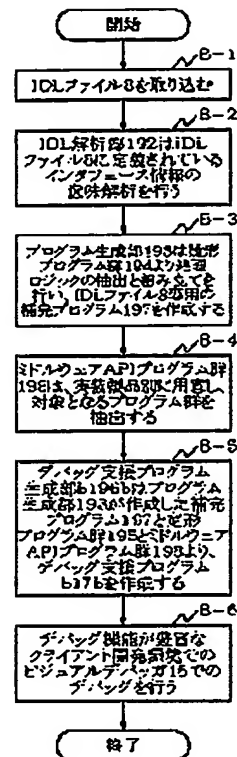
(8)

特開2001-256075

【図5】



【図6】



This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**